

SOA-CONFORM MODELING AS A HIGHLEVEL STANDARD FOR DISCRETE MODELLING AND SIMULATION

Thomas Wiedemann
University of Applied Science Dresden
Friedrich List Platz 1 Dresden, 01069
Germany
E-mail: wiedem@informatik.htw-dresden.de

ABSTRACT

Today, there is no universal standard for discrete simulation. Models, created with leading simulation tools can not be exchanged between the systems. This results in very high investments and maintenance costs for simulation studies and some additional problems with portability and performance in large simulations. This paper discusses a new approach by using Service Oriented Architectures (SOA). The integration of SOA-interfaces in simulation systems connects them to the common IT-infrastructure of ERP-systems, business process modeling systems and web-services. In addition, the SOA standards BPMN and BPEL are also capable for simulation modeling.

INTRODUCTION

The main algorithms and mathematical foundations of simulation systems are well defined and efficient (Wiedewitsch and Heusmann 1995). Nevertheless, the real application of simulation systems is still difficult (Kuljis and Paul 2000). Not more than 10% of all industrial firms use simulation tools by a number of reasons:

- Unlike the continuous simulation marketplace, there is no leading discrete simulation system. As a result, there is **no universal standard for discrete simulation**. Models can not be exchanged between the systems.
- As a result of the small market, the prices of the systems are very high. Typical prices of more than \$50,000 are too high for medium-sized firms.
- These problems indicate the need of a new strategy for the development of simulation tools. Like in the database software domain, **we need powerful standards for modeling and simulation**.

First steps were made by some Open-Source-simulation systems ideas, like DSO (Jacobs 2004) or SILK (Kilgore 2001). The main ideas of Open Source and the advantages for simulation tools are discussed in detail by Kilgore in the original paper outlining the OpenXML-project during the Winter Simulation Conference 2001 (Kilgore 2001).

But the mentioned systems were only focused on the simulation scenario. There is no option towards an open and universal standard linked to the common IT-

infrastructures, because the models are described and compiled with very specific JAVA-programs. A solution for this problem was presented by the author at recent Winter Simulation Conferences (Wie2002), by converting this JAVA program into a common, high-level XML-model description and a second conversion into an other simulation languages like SLX. The open question of this conversion process was the detailed definition of the high-level modeling language. Main requirements and possible solutions are discusses in the following chapters.

REQUIREMENTS FOR AN UNIVERSAL MODELLING LANGUAGE

Defining a standard is a very expansive and time consuming process. For this reason, existing standards and procedures should be used as much as possible. From the view of the front-end-user, the modeling systems should provide the following characteristics :

- **Readability** means, that the target audience like production controllers or managers should understand and manage the presentation by an easy and simple way, also like they manage the standard IT-processes.
- **Modularity** is related to readability in that a part time developer can make a change to the model description or replace an entire model module without having to understand or modify large amounts of the remaining model.
- **Extendibility** means that the modeling should include not only the basic simulation application, but also other tools like animation or optimization applications.
- **Distributed modeling** is required for the increasing number of supply-chain-simulation scenarios. The model description should support different simulation systems on different hardware and software platforms.

Concerning the IT-infrastructure, the universal modeling language should provide the following characteristics:

- **Acknowledged standard in the computer science community**. Former simulation interoperability standards like HLA are not known in the IT community !

- **Freedom of choice of tools** for managing the model description and the conversion processes to the simulation tools. Proprietary standards and closed conversion tools will not be accepted in the actual IT world, which is focused more and more on Open-source software licenses .

The mentioned requirements are very complex and sophisticated. Until today, there were a lot of attempts made, like the DEVS (Zeigler87) or using UML for describing simulation models – but no option was successful, concerning a wide use in practice and research.

SOA AND SIMULATION MODELING

Similar problems like in the simulation area were encountered also in the area of business process modeling and reengineering. Specific problems, which are very difficult for modeling concepts like UML are

- Description of highly parallel processes with delays and blocking effects between the processes
- Synchronous and asynchronous relationships between the processes
- human interactions with a high degree of uncertainty.

The concept of Service Oriented Architecture (SOA) (see (Woods 06) provides such an environment, where **loosely coupled services** (modules) interact with each other via an intermediary (Enterprise Services Bus) and the sequencing of service execution is controlled by a Service Orchestration Engine (SOE) (see fig.1).

Simulating processes require extremely flexible modeling facilities as described by M.Pidd and R. Bayer Castro in the Proceedings of the 1998 Winter Simulation “HIERARCHICAL MODULAR MODELLING IN DISCRETE SIMULATION” (Pidd 98). They describe the most common approach, in which control rules are hard-coded into the simulation models. A much better approach is to separate the control rules from the event code and represent them by external descriptions.

SOA and simulation modeling are both focused on the description of complex systems with random parameters, uncertainties and parallel running processes. In result, most existing concepts and tools of Service Oriented Architectures match the requirements of simulation modeling (Figure 1). Of course, the SOA concept is relatively new and the quality of the modeling depends on the specific tools and IT-infrastructures.

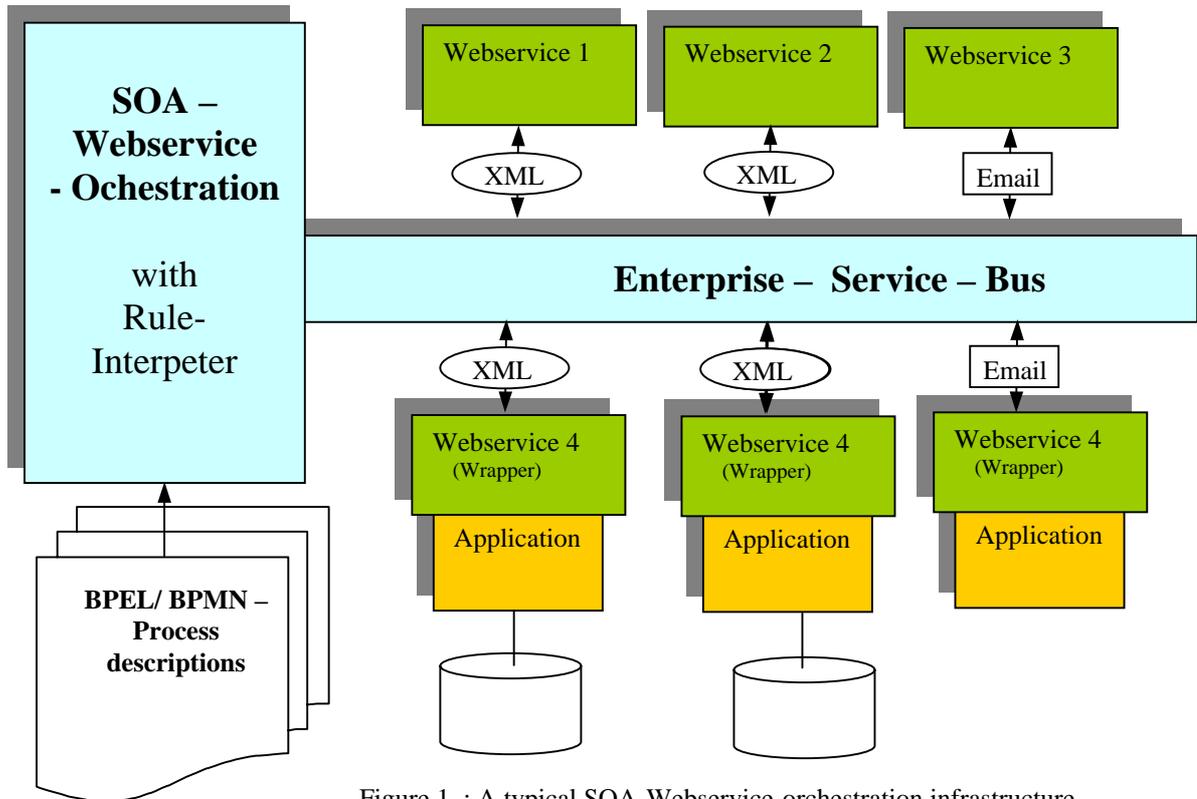


Figure 1 : A typical SOA-Webservice-orchestration infrastructure

MODELING WITH BPMN-RULES

Because BPEL is not very readable, a more powerful high-level modeling scheme is provided by the Business Process Modeling Notation (BPMN). Business Rules are described with activities (shown like boxes) and process interactions (shown with arrows) Fig. 2 shows a typical simulation scenario generated with the SOCON INTEGRITY designer. Any other tool can be used too. The model is readable also for non-simulation experts and managers.

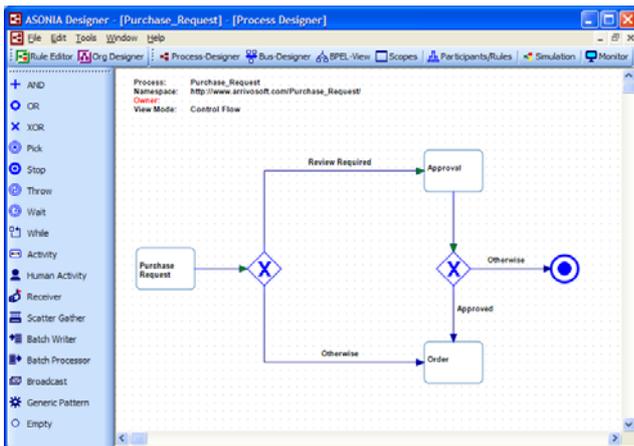


Figure 2 : The BPMN-designer (by SOCON INC)

From the BPMN model the corresponding BPEL model is generated automatically (see fig. 3). In most cases this BPEL is very large and complicated, so it is used only for very specific low level debugging or interface translations.

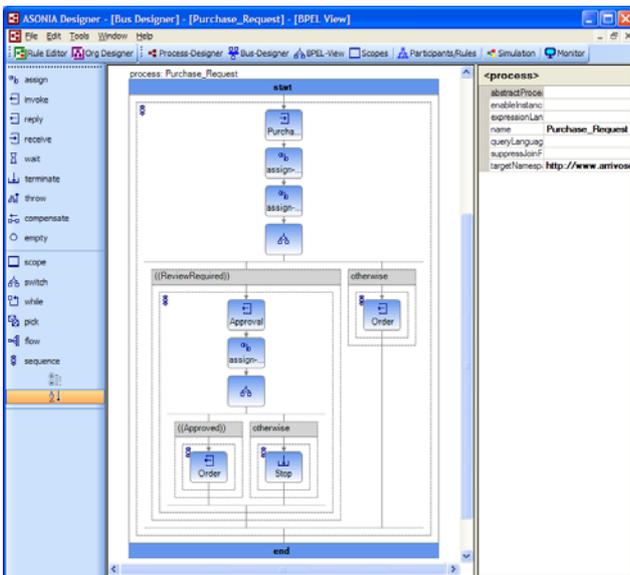


Figure 3 : The generated BPEL-model

COUPLING MODELS AND APPLICATIONS

In an SOA, a message intermediary, also known as an **Enterprise Service Bus (ESB)**, is used for the loosely coupled communication of services. It acts as an intermediary by sending, receiving and transforming standardized messages to and from services that are called by the business process. An Enterprise Service Bus refers to a software architecture construct that provides foundational services for more complex service-oriented architectures via an event-driven and XML-based messaging engine (the bus). An Enterprise Service Bus generally provides an abstraction layer on top of an Enterprise Messaging System which allows integration architects to exploit the value of messaging without writing code.

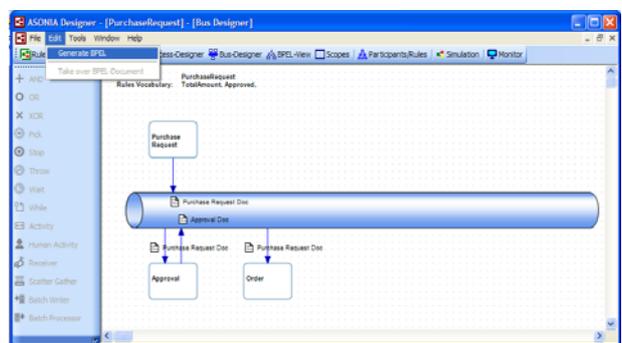


Figure 4 : The ESB-Bus

Additional tools, like a service registry in which physical web services are stored along with the their URL and the input/output schemas of the services' web methods, support the ESB. Process Logical Endpoints (or activities) can then be "assigned" to the web methods. This allows a fast change of configuration between different simulated and real endpoints.

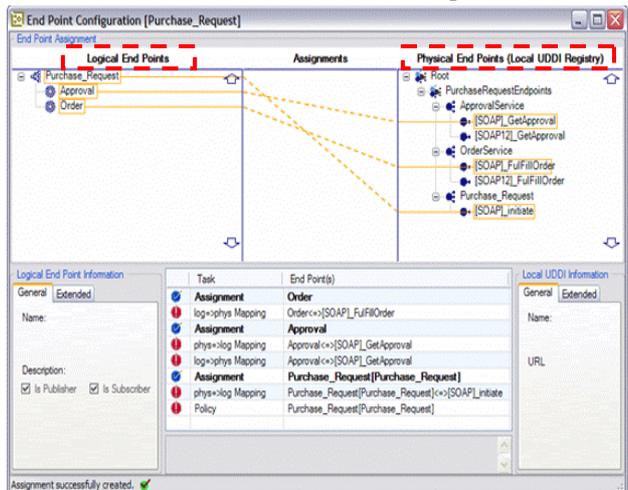


Figure 5 : Definition of logical endpoints

SIMULATION WITH SOA

The resulting BPEL-code of the model can be used in two ways:

- In a first step, the BPEL-code is converted to a simulation program or it is interpreted by a simulation system. Because the SOA tools provide already some simple simulation modules without stochastic calculation, the results of the real simulation run could be used for improvements and for large scale tests of the SOA-business model.
- If the simulation runs are finished and no problems are detected, the SOA-code is transferred to the real control software of the business process in the traditional manner.

By this dual use of the SOA model an old dream of simulation and business process developers become reality – the usage of **ONLY ONE MODEL** for the simulated and the real world. By the usage of the simulation model inside the SOA modeling process of real world software, the simulation community is not only a passive customer of SOA, but can provide very useful services in the SOA-world.

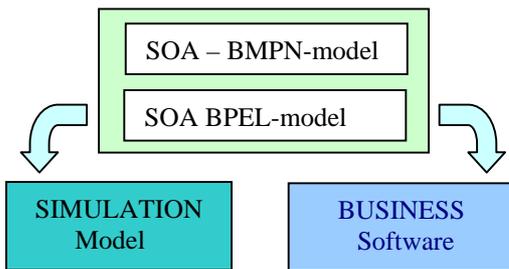


Figure 6: The dual use of the SOA model

ADVANTAGES OF SOA IN SIMULATION

The main advantage of SOA in simulation is the usage of SOA for simulation modeling by itself:

- Non-simulation IT professionals are already familiar with the SOA concepts and tools and they are ready to integrate simulation tools with SOA-interface into larger IT-environments.
- The SOA modeling concepts BPMN and BPEL are scientific proved and accepted by all major IT companies. There is no danger that this standard is not accepted or will not work in some cases.
- All defined rules in the BPMN model (see fig. 7) can be transferred or to the simulation execution layer or to the generator or to a real control software system. The common rule-repository eliminates errors by using different rules in the simulation and the real world.
- The SOA-Webservice-bus is common accepted by all major IT-software vendors like SAP, Oracle and Microsoft. By using SOA, the simulation world is connected to the remaining world without large efforts. This will decrease costs for development of interfaces many times and will increase the usage of simulation tools in testing and optimizing large software systems.
- Compared to the simulation specific HLA-standard for interoperability, the SOA-BUS is capable to connect ALL business software modules. In the future, there is no need of new interfaces, but all simulation and business software can use the same interfaces. It is also possible to change between simulated and real world software modules.

In conclusion, the SOA concept and tools will connect the simulation to the most sophisticated software technology for distributed real world business systems.

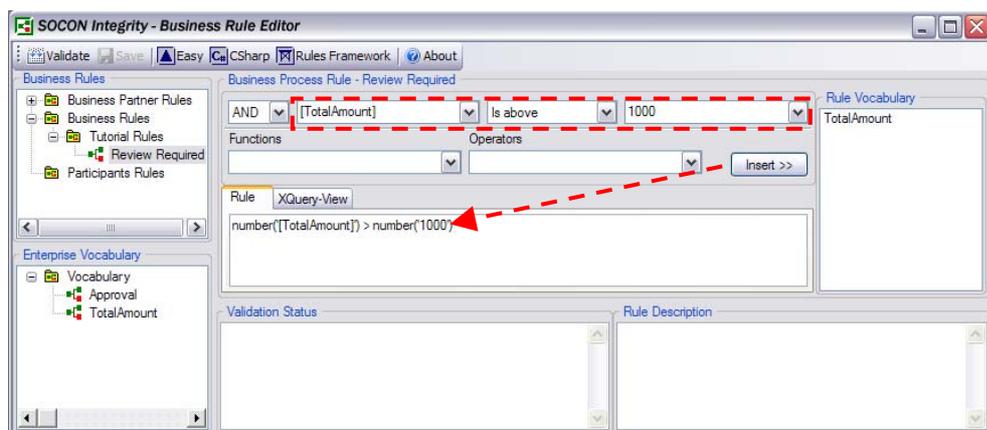


Figure 7: Definition of specific rules with the Business rule editor

NEW APPLICATIONS OF SIMULATION

The parallel run of simulated and real software modules could be a new killer application in IT development, because it allows to decrease the installation and testing cycle of large ERP systems, like SAP, from months to some weeks. Instead of waiting for real world hardware and software installations (like a 3 Mio. \$ Warehouse and warehouse software) the ERP system could be tested against a simulated warehouse. It is also easier to search a missing container inside a simulation model, instead in the real world. When the real world hardware is installed partly, already running modules could be included into the Simulated- and Real-Software-environment only by changing the end-point-configurations of the SOA-bus. If the whole system is completely tested, the fast switch-over to simulated hardware is still interesting for testing new software version of the ERP-system and for optimizing the production or storage process.

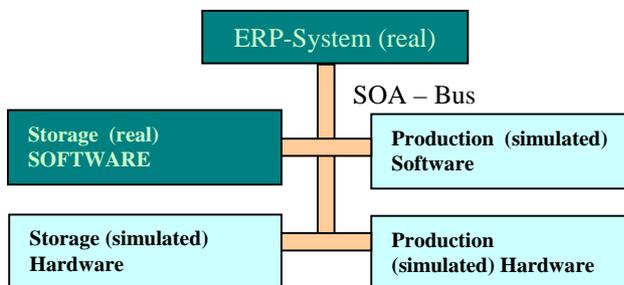


Fig. 8: Testing simulated and real Hardware/Software

SUMMARY

The presented SOA-based Simulation project could be a potentially beneficial evolution in the simulation software development model. The first advantage is the larger flexibility from the use of a standard commercial standard for modeling, which is also the first real standard in simulation since 30 years, if GPSS is encountered as an Quasi-Standards.

A specific advantage is the usage of an universal, language independent XML-description, which is compatible to all business software environments. With two sequential transformation processes a simulation model can be transferred between different platforms without manual changes. Together with the basic idea a universal and non-language-dependent simulation system can be provided in the future.

The actual state of the project is ongoing and further information is available at [SIMSOLUTION]. Its future development will provide the first time in simulation history a universal and open simulation system. Any interested simulation expert or user is invited by the authors for sharing his ideas, experience and cooperation inside the consortium.

REFERENCES

- Kilgore, R. A. 2001. Open source simulation modeling language (SML). In Proceedings of the 2001 Winter Simulation Conference, ed., B. Peters, J. Smith. Piscataway, NJ: 2001
- Kuljis, Jasna and Ray J. Paul, 2000: A Review of web based simulation: whiter we wander?, *Proceedings of the 2000 Winter Simulation Conference*, Orlando Florida, page 1872-1881
- Jacobs, Peter, 2004: The DSO Simulation System. *Proceedings of the European Simulation Symposium*, Budapest, Hungary, October 2004
- Pidd, M, and Castro 1998: Hierarchical modular modelling in discrete simulation. *Proceedings of the 1998 Winter Simulation Conference*, page 383-389
- SIMSOLUTION : Available online via <http://www.simsolution.net> [accessed April 1, 2007].
- Wiedemann, T., 2002. Next generation simulation environments founded on open source software and XML-based standard interfaces, *Proceedings of the 2002 Winter Simulation Conference*
- Wiedewitsch J.; and Heusmann J. 1995. "Future Directions of Modeling and Simulation in the Department of Defense", Proceedings of the SCSC'95, Ottawa, Ontario, Canada, July 34-26, 1995
- Zeigler, B.P. (1987). Hierarchical, Modular Discrete-Event Modelling in an Object-Oriented Environment. *Simulation*, vol. 50, pp. 219-230.
- Zeigler, B. P. (1990). *Object-oriented simulation with hierarchical, modular models*. Academic Press, Boston.
- Woods, D.; Mattern, T.: Enterprise SOA: Designing IT for Business Innovation. O'Reilly, (2006)

AUTHOR BIOGRAPHY

THOMAS WIEDEMANN is a professor at the Department of Computer Science at the University of Applied Science Dresden (HTWD). He has finished a study at the Technical University Sofia and a Ph.D. study at the Humboldt-University of Berlin. His research interests include simulation methodology, tools and environments in distributed simulation and manufacturing processes, intranet solutions and database applications.

Email : <wiedem@informatik.htw-dresden.de>